

For questions 1-3, consider a 4-color map coloring problem on the states above with colors red (R), blue (B), yellow (Y), and orange (O).

1. Show what happens to the variable domains when forward-checking is employed after each of the first two decisions:

a. CA is assigned B.

b. AZ is assigned O.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | CA | NV | UT | AZ | CO | NM | TX |
| Initial Domains | R B Y O | R B Y O | R B Y O | R B Y O | R B Y O | R B Y O | R B Y O |
| CA is assigned B |  | R Y O | R B Y O | R Y O | R B Y O | R B Y O | R B Y O |
| AZ is assigned O |  | R Y | R B Y |  | R B Y O | R B Y | R B Y O |

a. NV and AZ remove ‘B’ in their domains.

b. NV, UT, and NM remove ‘O’ in their domains

2. Repeat problem 1 using maintaining arc consistency instead of forward checking.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | CA | NV | UT | AZ | CO | NM | TX |
| Initial Domains | R B Y O | R B Y O | R B Y O | R B Y O | R B Y O | R B Y O | R B Y O |
| CA is assigned B |  | R Y O | R B Y O | R Y O | R B Y O | R B Y O | R B Y O |
| AZ is assigned O |  | R Y | R B Y |  | R B Y O | R B Y | R B Y O |

a. Check (CA, NV)🡪(CA, AZ)🡪(AZ, NM)🡪 (AZ, UT) 🡪 (AZ, NV) 🡪 (NV, UT): NV and AZ remove ‘B’ in their domains.

b. Check (AZ, NV)🡪(AZ, UT)🡪(AZ, NM)🡪 (NV, UT) 🡪 (UT, CO) 🡪 (NM, CO)🡪(NM,TX): NV, UT, and NM remove ‘O’ in their domains

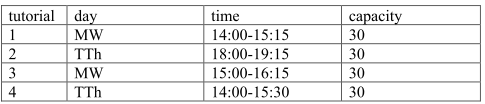
3. Show how the conflict sets would be updated for these two moves if we were using

conflict-directed backjumping.

{}🡨 Conflict set ; ⃝🡨 Assigned

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | CA | NV | UT | AZ | CO | NM | TX |
| Conflict set |  |  |  |  |  |  |  |
| CA is assigned B |  | { CA= R } |  | { CA= R } |  |  |  |
| AZ is assigned O |  | { CA= R, AZ=O } | { AZ=O } |  |  | { AZ=O } |  |

4. Four courses are available:



Suppose students have the following set of simplified constraints:

i. Students cannot be enrolled in overlapping courses.

ii. Students cannot enroll in classes on days that they work.

iii. A student may not enroll in a course that does not have space.

a. Write a specification for this constraint satisfaction problem.

Variables:

Divide a student’s timetable into four parts.

1. MW 14:00~15:15 🡪 SiMW1 (A period, 14:00 to 15:15, of Student i on Monday and Wednesday)

2. MW 15:00~16:15 🡪 SiMW2 (A period, 15:00 to 16:15, of Student i on Monday and Wednesday)

3. TTh 14:00~15:30 🡪 SiTTh2 (A period, 14:00 to 15:30, of Student i on Tuesday and Thursday)

4. TTh 18:00~19:15 🡪 SiTTh2 (A period, 18:00 to 19:15, of Student i on Tuesday and Thursday)

One student has 4 periods, so we have i\*4 variable.

For example, Variables= {S1MW1, S1MW2, S1TTh1, S1TTh2, S2MW1, S2MW2, S2TTh1, S2TTh2, …}

Domain:

Four Courses, C1, C2, C3 and C4. One Empty, E, and a Work, W.

C1: A course whose Tutorial is 1 E: The period is free.

C2: A course whose Tutorial is 2 W: Students work either days.

C3: A course whose Tutorial is 3

C4: A course whose Tutorial is 4

For example, S1MW1 = W means S1 works either Monday or Wednesday.

S2MW1 = C1 means S2 take a course C1.

Constraints:

1. MW1 and MW2 cannot take course simultaneously. If MW1 = C1, the other must be E.

2. TTh1 and TTh2 cannot take course simultaneously. If TTh1 = C1, the other must be E

3. If MW­x is W then the other must be W as well. 🡪 MW1 = W ↔MW2 = W

4. If TTh­x is W then the other must be W as well. 🡪 TTh1 = W ↔ TTh2 = W

5. The number of Cx cannot be more than 30. E.g. Cannot be more than 30 students taking C1.

b. Show how encapsulation could be used to create a binarize the constraint between

multiple students the enrollment limit.

i. Students cannot be enrolled in overlapping courses. & ii. Students cannot enroll in classes on days that they work.

Combine MW1 and WM2 into a set and TTh1 and TTh2 into a set. The elements in a set cannot be two Ci. Also, if a element is W, the other must be W too.

E.g. Ui,ii = [ { S1MW1= C1, S1MW2= C2 }, { S1MW1= C1, S1MW2= E }, { S1TTh1= C3, S1TTh2= W }, { S1TTh1= E, S1TTh2= C4 }, … ]

{ S1MW1= C1, S1MW2= C2 } = False

{ S1MW1= C1, S1MW2= E } = True

{ S1TTh1= C3, S1TTh2= W } = False

{ S1TTh1= E, S1TTh2= C4 } = True

iii. A student may not enroll in a course that does not have space:

We can combine all SxMW1 to be a set and count how many C1. If the number of C1 is larger than 30 then it conflicts. Following this rule, we can combine different MW and TTh to be sets respectively.

U = [ {S1 = C1, S2 =C1, S3 = C1, S4 = E, S5 =E …, Si =W…, Sj =C1…}, {S1 = C2, S2 =C2, S3 = C2, S4 = E, S5 =E …, Si =W…, Sj =C2…}, {S1 = C3, S2 =C3, S3 = C3, S4 = E, S5 =E …, Si =W…, Sj =C3…}, {S1 = C4, S2 =C4, S3 = C4, S4 = E, S5 =E …, Si =W…, Sj =C4…}].